SkyHash: a Hash Opinion Dynamics Model

Houwu Chen and Jiwu Shu*
Department of Computer Science and Technology
Tsinghua University
{chenhw11@mails., shujw@}tsinghua.edu.cn

Abstract—For consensus on hash opinions in P2P networks, an opinion dynamics model named SkyHash is proposed. The model consists of a bit layer and a hash layer, where for each round of a node, the bit layer is to determine each bit of a pseudo hash, and the hash layer is to choose a hash opinion with minimum Hamming distance to the pseudo hash. Impacts of network size, node degree, hash size and initial hash density on the convergence performance of the model are studied by simulations, which also shows that for a homogeneous network with 20000 nodes, average node degree is 33 as well as 256-bit hash size, consensus can be reached within 14 rounds. Denial of service (DoS) attack is pointed out and a DoS-proof extension for the model is developed. Experiments on the SNAP dataset of the Wikipedia who-votes-on-whom network demonstrate that with reasonable latency assumption, the DoS-proof extended model produces consensus in 45 seconds, and tolerates DoS attack committed by 7% random nodes or 0.9% top influential nodes at the cost of 50% throughput reduction. To the best of our knowledge, it’s the first hash opinion dynamics model.

I. INTRODUCTION

Opinion dynamics is a field utilizing computational tools or mathematical-and-physical models to explore the dynamical processes of the diffusion and evolution of opinions in a society, where individuals shape their opinions based on the opinions they receive from a subset of the society [1]. Existing opinion dynamics models are for binary opinion (e.g. majority rule, voter and Sznajd), continuous opinion (e.g. Deffuant and Hegselmann-Krause) or vector opinion (e.g. Axelrod) [2]. However, for consensus on committing arbitrary transactions identified by their hashes in P2P networks (e.g. cryptocurrency [3]), when opinion dynamics is applied [4], a hash opinion dynamics model is required. Hash opinion should not be treated as continuous opinion by regarding each hash as a big number, or vector opinion by regarding each bit of a hash as a standalone binary, because two attributes should be reserved: one is that each hash is on a par with another hash, and they can’t be compared on magnitude; the other is that each hash is calculated from its corresponding data rather than simply combined by individual bits. Those attributes make hash opinion dynamics a unique problem.

In this paper, we proposed SkyHash, a hash opinion dynamics model for P2P networks constructed by trust relationships. Each node in such a P2P network shapes its opinion by rounds, in each round it receives opinions from its followees, applies the SkyHash model to determine its new opinion, and then shares the new opinion to its followers. The SkyHash model consists of a bit layer and a hash layer, where for each round of a node, the bit layer is to determine each bit of a pseudo hash, and the hash layer is to choose a hash opinion with minimum Hamming distance to the pseudo hash. Simulations indicate that the round needed for full convergence linearly increases with \( \log D N \), where \( D \) is average node degree and \( N \) is network size. Simulations further unfold performance decreases in heterogeneous network comparing to homogeneous network with same parameters. Simulations also show that convergence performance increases with hash size as well as initial hash density. For a homogeneous network with even 20000 nodes, average degree is 33 and 256-bit hash size, consensus can be reached within 14 rounds in simulations. We also discovered denial of service (DoS) attack and developed a DoS-proof extension for the model. Key points of the implementation are introduced especially on the asynchronous aspect. Experiments on the SNAP dataset of the Wikipedia who-votes-on-whom network [6] demonstrates that with reasonable latencies assumption, the DoS-proof extended model produces consensus in 45 seconds, and tolerates DoS attack committed by 7% random nodes or 0.9% top influential nodes at the cost of 50% reduction of throughput. To the best of our knowledge, it’s the first hash opinion dynamics model.

II. PROBLEM AND DATASETS

P2P networks are assumed to be constructed by trust relationships. As shown in Fig. 1 when node \( A \) trust a node \( E_i \), \( E_i \) is a followee of node \( A \) whereas \( A \) is a follower of \( E_i \). Opinions flow from followees to followers unidirectionally. In this way, the network can be abstracted to a directed graph where each trust relationship is a directed edge.

During an opinion dynamics process, a node \( A \) shapes its opinion **by rounds** as shown in Fig. 2. In each round, \( A \) receives opinions from its followees, and those opinions together with the opinion of \( A \) itself form an **opinion set** as shown in Fig. 1. \( A \) then applies the opinion dynamics model to determine its new opinion according to the opinion set, and...
tell its new opinion to its followers by share the new opinion in a sharing group(a.k.a. swarm) consists of its followers as shown in Fig. 1. After that, A enters the next round.

Fig. 2. Node State

This paper focuses on the model to be applied in the the greyed state in Fig. 1 to produce the convergence of hash opinions in the whole network. We use the terms convergence and consensus interchangeably in this paper.

Our strategy is to analyze the opinion dynamics model in a synchronous process where each round for each node takes exact 1 unit of time, and opinions determinately flow from followees to followers taking no time. We then implement the model with practical asynchronous time assumption. Such an initial synchronous phase is sometimes called a conciliator [5].

Our model is evaluated on the SNAP dataset of Wikipedia who-votes-on-whom network [6], which presents trust relationships in the form of votes for administration and is named the wiki dataset in this paper. We also impose a constraint which can be enforced in P2P client of each well-behaved node that indegree > = 10, thus all nodes with followees less than 10 are removed. Parameters of the result network is shown in Table I and the cumulative distributions of indegrees and outdegrees are shown in Fig. 3.

Fig. 3. Degree distribution of the wiki dataset

To reveal the impact of network size, we also run simulations upon various homogeneous networks with size of 100, 1000, 5000 and 20000 nodes separately, where each node has the same indegree as the average degree of the wiki dataset, and connects to each other randomly. Those datasets are named uniform-100, uniform–1k, uniform–5k and uniform–20k respectively.

III. The SkyHash Model

The hash opinion dynamics model can be abstracted to a function F applied to an opinion set H to produce a hash value Hx as shown in Eq. (1), where the opinion set H = {H0, H1, H2, ⋯ Hn}, H0 is the hash opinion of the current node, and Hi for i ∈ [1, n] is the hash opinion of the i-th of n followees.

\[ H_x = F(H) \]  

(1)

The SkyHash model we proposed consists of a bit layer and a hash layer as shown in Fig. 4.

A. Bit Layer

Bit layer is applied at each bit position j for j ∈ [1, k], where k is the hash size. First, for each hash Hi, bit at position j of Hi marked as bij is extracted. Then a value bij is determined on the bit set \{b0j, b1j, ⋯ bij\} according to the Sky bit layer model [4], which is a mix of a majority rule model and a simulated annealing model. Bit layer model can be implemented as the function BITSKY in Fig. 5 where BITMR is the majority rule model which mainly picks the majority opinion, and BITSA is the simulated annealing model which mainly picks an opinion with probability corresponding to the density of the opinion.

B. Hash Layer

After applying the bit layer at each bit position j to determine a bit bij, a pseudo hash P can be constructed as \[ P = b0b1b2⋯bk \]. This layer then computes the Hamming distance between each Hi and P, and picks from H the hash Hx with minimum hamming distance, as shown by function HASH in Fig. 6.

---

**Table I**

<table>
<thead>
<tr>
<th>Name</th>
<th>Wiki</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes Counts</td>
<td>998</td>
</tr>
<tr>
<td>Average Degree</td>
<td>33.33</td>
</tr>
<tr>
<td>Diameter</td>
<td>5</td>
</tr>
<tr>
<td>Average Path Length</td>
<td>2.34</td>
</tr>
<tr>
<td>Density</td>
<td>0.033</td>
</tr>
<tr>
<td>Average Clustering Coefficient</td>
<td>0.183</td>
</tr>
<tr>
<td>Eigenvector Centrality Sum Change</td>
<td>0.029</td>
</tr>
</tbody>
</table>

**Datasets parameters**

---

**Fig. 4. The SkyHash model**
C. Simulations

Impacts of various factors on convergence performance are revealed by simulation results exhibited in Fig. 7.

![Simulation results](image)

**Fig. 7. Simulation on various factors**

1. **function** BITSKY([b$_{0j}$, b$_{1j}$, ..., b$_{ij}$])
2. $n_0 \leftarrow$ count 0 in \{b$_{0j}$, b$_{1j}$, ..., b$_{ij}$\}
3. $n_1 \leftarrow$ count 1 in \{b$_{0j}$, b$_{1j}$, ..., b$_{ij}$\}
4. return randomly pick in \{BITMR($n_0$, n$_1$), BITSA($n_0$, n$_1$)\}
5. **function** BITMR($n_0$, n$_1$)
6. if $n_0 > n_1$ then
7. return 0
8. else if $n_1 > n_0$ then
9. return 1
10. return randomly pick in \{0, 1\}
11. **function** BITSA($n_0$, n$_1$)
12. $n \leftarrow n_0 + n_1$
13. if $n_0 > n \times 0.8$ then
14. return 0
15. else if $n_1 > n \times 0.8$ then
16. return 1
17. test $\leftarrow$ randomly pick in \{0, n\}
18. if test $< n_0$ then
19. return 0
20. else if test $> n_0$ then:
21. return 1
22. return randomly pick in \{0, 1\}

**Fig. 5. Bit layer algorithm**

1. **function** HASH([H$_0$, H$_1$, H$_2$, ..., H$_n$], P)
2. min$_d \leftarrow k + 1$
3. **for all** H$_i$ in \{H$_0$, H$_1$, H$_2$, ..., H$_n$\} **do**
4. test $\leftarrow$ bitwisely apply XOR between H$_i$ and P
5. d $\leftarrow$ count 1 in all bits of test
6. if d $< \text{min}_d$ then
7. H$_x$ = H$_i$
8. min$_d$ = d
9. return H$_x$

**Fig. 6. Hash layer algorithm**

Fig. [f(a)] demonstrates the impact of network size on convergence performance. Simulations are executed on various datasets where hash size is 256-bit, and initially each node hold an random hash opinion. The horizontal axis of Fig. 7(a) is the round of the network, where all nodes are always at the same round as stated in section II. The vertical axis of Fig. 7(a) is the density of the top hash which is the hash opinion hold by the most number of nodes in the whole network. The figure unfolds the following results:

- For homogeneous networks with the same average degree(e.g. all the uniform-* datasets), round needed for full consensus increases with network size, however, it increases slowly along with network size, e.g., when network size increase from 100 to 20000, round to converge needed only increases from 6 to 14.
- For heterogeneous network, e.g, the wiki dataset, convergence performance degrades remarkably comparing to the homogeneous network with same size and average node
degree. Similar result is also observed in existing studies which shows that community strength in a heterogeneous network impacts the performance significantly [7], [8].

- Convergence increases quickly at the intermediate rounds for all datasets, however, for slower simulations, it takes more time to escape from disorder when convergence is near 0 and to full order when convergence is near 1.

Fig. 7(d) demonstrates the impact of average node degree on convergence performance. Simulations are executed on homogeneous networks with 1000 nodes, but with various average node degree respectively. Also, each node holds a random hash opinion initially, and the hash size is 256-bit. The horizontal and vertical axes are same with Fig. 7(a). The figure shows that for fixed sized homogeneous networks, convergence performance increases with degree.

Data in Fig. 7(a) and Fig. 7(b) for homogeneous networks is presented again in Fig. 7(c), where horizontal axis is $log_D N$, $D$ is average node degree and $N$ is network size. The figure shows that the round needed for full consensus linearly increases with $log_D N$.

Fig. 7(d) demonstrates the impact of hash size on convergence performance. Simulations are executed on the uniform–1k dataset with various hash sizes while each node hold a random hash opinion initially. The horizontal and vertical axes are same with Fig. 7(a). The figure shows convergence performance increases with hash size for a given dataset.

Fig. 7(c) demonstrates the impact of initial hash density on convergence. Simulations are executed on the uniform–1k dataset with 256-bit hash size and various numbers of initial opinions hold by all the nodes evenly. For example, for the 2 initial opinions case, there are two hash opinions in the whole network, and each hash opinion is hold by half number of the nodes. The figure shows that for a given dataset, convergence performance decrease with initial opinions count.

IV. THE DO$S$-PROOF EXTENSION

A. Denial of Service Attack

Nodes in a P2P network may be ill-behaved, and they do not obey the proposed model or even collude with other nodes. For big hash size such as 256-bit, it is impossible at present to elaborate data so that its hash is same as a given value (a.k.a. hash collision). As a result, behaviors of ill-behaved nodes are constrained or they will be identified on discrepancy of the hash and the corresponding data. However, existing studies show that if ill-behaved nodes collude together to keep telling other nodes a fixed opinion disregarding the opinions of their followees, even a small number of the such nodes may control the opinion of the whole network, and such nodes are usually called stubborn agents or committed minorities [9], [10].

Simulations (not presented in this paper due to capacity) reveal that even 0.5% of such nodes can prevent the whole network to agree at well hashes proposed by well-behaved nodes, and only ill hashes proposed by ill-behaved nodes are agreed at. Such a case is named denial of service (DoS) attack.

B. The Extended Model

Based on the observation that the higher density of well top opinion, the stronger to resist attack [4], we proposed a DoS-proof extension consisting of two phases: a reverse phase and a normal phase. The extension can be implemented as algorithm described in Fig. 9 where HASH (the normal phase) is exactly the one in Fig. 6 and RHASH (the reverse phase) is almost same as HASH except it picks the hash $H_x$ with the maximum Hamming distance.

1: function HASHDOS($\{H_0, H_1, H_2, \cdots, H_n\}$, $P$, $r$) $\triangleright$ $r$ is the round number
2: if $r < R$ then $\triangleright$ $R$ is a given threshold
3: return RHASH($\{H_0, H_1, H_2, \cdots, H_n\}$, $P$)
4: return HASH($\{H_0, H_1, H_2, \cdots, H_n\}$, $P$)

Fig. 9. DoS-proof extension algorithm

C. Simulations

With round threshold $R = 15$, simulations on the uniform–1k dataset with DoS attack from 11%, 15% and 20% nodes respectively are shown in Fig. 8. The horizontal and vertical axes are same with Fig. 7(a). Green lines are the cases that for each case the network succeeds to resist the DoS attack, where all well-behaved nodes agrees at a well hash. Red lines are the cases that for each case the network fails to resist the DoS attack, where all well-behaved nodes agrees at the ill hash. Solid lines are the density of the ill hash, and dashed lines are the density of the top hash (may be well hash or ill hash), which is the hash opinion hold by the most number of nodes in the whole network.

The network is able to survive DoS attack by less than 15% nodes, but 50% of the runs agree on ill hashes, thus the throughput will decrease by 50%. Fig. 8(a) and Fig. 8(b) demonstrate the oscillation of the density of the ill hash, and show that the heavier the attack the smaller range the oscillation, until the oscillation is insignificant and in all runs the network always agrees on the ill hash as in Fig. 8(c).

V. IMPLEMENTATION AND EXPERIMENTS

A. Implementation

To implement the model, each node publishes a public key as its identity, and each opinion the node shares is signed by its private key. A sharing group (known as a “swarm”) is identified by the public key of a node, and its followers join the swarm by finding the public key in a distributed hash table (DHT).

For each node, a failure detector is utilized to deal with the FLP impossibility problem in asynchronous system [11], [12]. As shown in Fig. 10, the failure detector maintains an active followees list as well as a suspect followees list (as state $E$ and $G$). A followee is initially in the active list, it is moved to the suspects list (as action in $G \rightarrow H$) if no up-to-date opinion is received when timeout (as $C \rightarrow G$), and moved back to active list if a new up-to-date opinion is received (as $E \rightarrow F$).
Each opinion is attached with a flag in \{deciding, decided, confused\} denoting the current status of the corresponding node. A node starts with deciding (as action in $A \rightarrow B$), and keep this flag (as action in $I \rightarrow B$) until a given round threshold is reached (as condition in $H \rightarrow J$) and then the node is terminating the current consensus process (as state $J$). It then share its opinion with flag decided if over $2/3$ of its active followees share the same opinion (as $J \rightarrow K$ with condition safe), or share its opinion with flag confused (as $J \rightarrow K$ with condition unsafe).

An opinion filter (as state $D$) is utilized by each node to check whether an opinion received is up-to-date or not. When an opinion is shared by a node, it also attaches the round number. An opinion is considered to be up-to-date if the current round of the node finishes (as state $H$), according to the relationship of the node's round number and the given round threshold (as $H \rightarrow J$ or $H \rightarrow I$), the node either terminates (as state $J$) or applies the model (as state $I$), share new opinion (as $I \rightarrow B$), and enters the new round (as state $B$).

### B. Experiments

According to existing studies, latencies between peers in DHT are mostly between 50 to 1000 ms [13]. Our experiments employ a simply latency model that the times to deliver opinions conforms gauss distribution of ($\mu = 500, \sigma = 500$) with lower cutoff of 50 and no upper cutoff which means an opinion may be lost in a small probability. Additionally, $timeout = 2000$ and round threshold $R = 15$.

Fig. 11 exhibits the experiment results on the wiki dataset. The horizontal axis of each sub figure is time in unit of millisecond. The vertical axis of each sub figure is the density of the top hash which is the hash opinion hold by the most number of nodes in the whole network. Green lines are the cases that for each case the network succeeds to resist the DoS attack, where all well-behaved nodes agrees at a well hash. Red lines are the cases that for each case the network fails to resist the DoS attack, where all well-behaved nodes agrees at the ill hash. Solid lines are the density of the ill hash, and dashed lines are the density of the top hash (may be well hash or ill hash), which is the hash opinion hold by the most number of nodes in the whole network. In each sub figure, solid line is for all well-behaved nodes, while dashed line is for all nodes with opinions decided only.

The wiki dataset can survive DoS attack committed by 7% random nodes (as shown in Fig. 11(b)) or 0.9% top influential nodes defined as the top 0.9% nodes by sorting all nodes in descendant order on the count of a node’s followees (as shown in Fig. 11(c)). However, the throughput will decrease 50% even when the network survives. In all the cases that the network survives, well-behaved nodes can always reach consensus within 45 seconds without well-behaved nodes agree at different values, while under DoS attack by 7% random nodes, 1.5% nodes are confused, and under DoS attack by 0.9% top influential nodes, 4% nodes are confused.
VI. RELATED WORK

Systematization of knowledge on opinion dynamics is introduced based on the viewpoint of statistical physics, and popular models including the voter model, majority rule model, models based on social impact theory, the Šnajd model, bounded confidence models and other models are briefly discussed in [11, 2] gives a multidisciplinary review on opinion dynamics, and brief comparison of the various models is also given by categories. [4] is the first work to bring opinion dynamics into P2P network. However, no hash opinion dynamics model is introduced at present.

As the source of DoS attack exhibited in this paper, the presence of stubborn agents (another name is committed minorities) in opinion dynamics is also studied in [9, 10, 4], but their primary focus is the impact of those stubborn agents thus no countermeasure is provided.

Similar to our observation on performance decrease in heterogeneous network comparing to homogeneous network with same parameters, [2] reveals that the convergence time decays exponentially with increasing community strength. [14] points out that strongly coupled nodes within the same community synchronizing their opinions faster than other nodes. [15] further indicates a transition at a value of the interconnectivity parameter, and communities reach consensus or opposite opinions when above or below the value respectively.

VII. CONCLUSION AND FUTURE WORK

To sum up, the SkyHash model can produce consensus in P2P networks for hash opinions, and the convergence performance linearly decreases with $\log_D N(D)$ (average node degree, $N$ is network size), while increases with hash size and initial hash density. Convergence performance also decreases on heterogeneous networks with communities. The DoS-proof extension of the model is effective to resist DoS attack at the cost of throughput reduction. Experiments show that the model enables a real world P2P network to reach consensus within 45 seconds. To the best of our knowledge, it’s the first hash opinion dynamics model.

To circumvent the impact of communities on the convergence performance, we are developing a ground truth community aware opinion dynamics model which leverages known member inclusion information of communities (e.g. chat rooms). Preliminary simulations of the model exhibit vast improvement on convergence performance.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 61433008), the National High Technology Research and Development Program of China (Grant No. 2013AA013201), and Project of science and technology of Beijing City (Grant No. D151100000815003).

REFERENCES
